Department of Electrical Engineering
Indian Institute of Technology, Bombay

EE 691: RnD Project
Multimodal Density Estimation from random linear compressive projections

Jian Vora, 170100026

Faculty Mentor: Prof. Vivek Borkar

December, 2020

Contents

1. Introduction

Given a set of N random variables $X_1, X_2, ..., X_N$, estimating the joint density $p(X_1, X_2, ..., X_N)$ is a fundamental problem in many fields such as statistics and machine learning (1). This probabilistic interpretation can help us make many inferences to help us aid take better decisions for the problem in hand. For a simple example, take X_1 as the time taken to drive to a particular destination and X_i, $2 \leq i \leq N$ denote a scalar number indicating the traffic on the $N-1$ streets. We would want to make predictions such as the minimum time needed for reaching the destination given we somehow know the traffic profiles of all the streets, i.e. more formally, we would want to find the following : min $p(X_1|X_2, X_3, ..., X_N)$ where the minimum is over the choice of routes. Capturing the joint density is helpful as we can infer a variety of queries of such form easily. Most the work done can be classified in a few categories, namely :

1. If the realisations of these random variables are discrete, then standard histograms would work, but it requires a large number of samples (exponential in N) to make the estimate of the joint density close to the actual density and is clearly not scalable with large N

2. Bayesian networks or Markov random fields which explicitly model some (in)dependencies between the set of random variables and hence scalable to larger datasets however require a lot of approximation techniques like variational inference (2) from a variety of common queries like evidence, marginal, conditional queries (3).

3. A Gaussian mixture model is among the simplest ways of a capturing the probability density by modelling it as a mixture of gaussians. These are like kernel density estimates, but with a small number of components (rather than one component per data point)(4). This is indeed shown to be a universal approximater, i.e. with enough number of components, it can approximate any density closely. This however is not very practical to use because the number of components increases very rapidly for even commonly occurring densities.

Given the relevance of the problem of density estimation, we would want to find the density of the N variables while still saving us from the curse of dimensionality. We aim to answer a few important questions regarding this topic. Can we do some transformation of the data to try and fit some simple estimators even though the density in the higher dimension is much more generic and expressive? If yes, what can this transformation be? These are some of the questions which this report tries to answer.

2. Preliminaries

2.1 Mixture of Log-Concave Densities

A log-concave density, as the name suggests, is defined as a density whose logarithm results in a concave function. Although these seem to be restrictive, these indeed cover a variety of commonly encountered densities such as a Gaussian, Laplacian, Beta and even a uniform distribution defined over a convex set. It is reasonable to assume that a large variety of data is generated following a mixture of log-concave densities. It is reasonable to assume data drawn from a mixture of such rich distributions which can also model heavy tails as opposed to a naive Gaussian mixture model which in fact is a special case of the above assumption. It is evident that we cannot directly learn such a density estimator in the high dimensional data because we do not know which mixture component follows which form of parametric distribution which is needed to learn general mixtures using standard techniques like Expectation Maximization.

More formally, our model can be formulated as follows: Consider a random vector $X \in \mathbb{R}^N$ which follows a mixture of log-concave densities $f_i(x)$ with K (finite) components subject to $\sum_i w_i = 1$, $w_i \geq 0$ and each f_i being log concave

$$f_X(x) = \sum_{i=1}^{i=K} w_i f_i(x)$$

This is model which shall be used for all the subsequent sections of the report whenever we refer to the density of the high dimensional data.

2.2 Random Linear Projections

For a given vector $X \in \mathbb{R}^N$, we define the following operation as random linear projections:

$$Y = \Phi X$$

where $\Phi \in \mathbb{R}^{M \times N}$ is composed of entries drawn i.i.d from a standard normal Gaussian or a Bernoulli $\{-1, 1\}$ distribution with the columns scaled accordingly. If $M < N$, then we call the projection as compressive which is the regime we will consider in this report. Such a forward model is commonly used for compressive sensing of signals and random projections have also been shown to do dimensionality reduction much more computationally efficiently as compared to some other expensive methods like PCA. Most work on random projections want the matrix Φ to follow the restricted isometry property, which in other words means that Φ is *almost* orthogonal which helps in preserving distances between the points even after projection.

2.3 Johnson Lindenstrauss Lemma

The lemma has uses in compressed sensing, manifold learning, and dimensionality reduction. Much of the data stored and manipulated on computers, including text and images, can be represented as points in a high-dimensional space. However, the essential algorithms for working with such data tend to become bogged down very quickly as dimension increases. It is therefore desirable to reduce the dimensionality of the data that preserves its relevant structure.

Formally, given $0 < \epsilon < 1$, a set X of m points in $\mathbb{R}^N$ and a number $n > 8ln(m)/\epsilon^2$, there exists a linear map $f : \mathbb{R}^N \mapsto \mathbb{R}^n$ such that

$$(1-\epsilon)||u-v||^2 \leq ||f(u)-f(v)||^2 \leq (1+\epsilon)||u-v||^2$$

for all $u, v \in X$(5). Note that the dimension of the projected subspace n depends only on m which is the number of datapoints and not the actual dimension of the data N. Let f be a linear projection matrix Φ, and taking $v = 0$, we can restate the JL Lemma in the following manner:

$$(1+\epsilon)^{-1}\|\Phi u\|^2 \leq \|u\|^2 \leq (1-\epsilon)^{-1}\|\Phi u\|^2$$

with the probability of

$$Pr(\|\Phi u\|_2^2 \in [(1-\epsilon)\|u\|_2^2, (1+\epsilon)\|u\|_2^2]) \geq 1 - n^{-2}$$

3. Random Linear Projections of Log-Concave Densities

3.1 Results for Isotropic Random Vectors

A random vector $X \in \mathbb{R}^n$ is said to be isotropic if $\mathbb{E}(X) = 0$ and $\mathrm{Cov}(X) = I_n$ where I_n is the $n \times n$ identity matrix. The Grassman manifold $G_{n,l}$ of all $l-$dimensional subspaces of $\mathbb{R}^n$ carries a unique rotationally-invariant probability measure $\mu_{n,l}$. Whenever we say that E is a random -dimensional subspace in $\mathbb{R}^n$, we relate to the above probability measure $\mu_{n,l}$. Under the additional assumption that the random vector X is isotropic, the subspace E for which $\mathrm{Proj}_E(X)$ is approximately Gaussian may be chosen at random and this holds with high probability. Formally the lemma is as follows:

Lemma 1: *Let X be an isotropic random vector $\in \mathbb{R}^n$ with a log-concave density. There exists a subset $\Theta \subseteq S^{n-1}$, with $\sigma_{n-1}(\Theta) \geq 1 - exp(-\sqrt{n})$ such that for any $\theta \in \Theta$ and any measurable set $A \subseteq \mathbb{R}$,*

$$|\mathbb{P}(X.\theta \in A) - \frac{1}{\sqrt{2\pi}} \int_A \exp(-s^2/2) ds| \leq \frac{C}{n^\alpha}$$

where $C, \alpha > 0$ are universal constants. (6)

Thus a random linear projection of an isotropic vector drawn from a log-concave density is close to a Gaussian in a total variation sense with high probability. In another work by Eldan and Klartag, the following statement was made stronger by showing the pointwise closeness of the projected density with an isotropic gaussian. More specifically,

Lemma 2: *Let X be an isotropic random vector in $\mathbb{R}^n$ with a log-concave density. Let $1 \leq l \leq n^{c_1}$ be an integer. Then there exists a subset $\epsilon \subseteq G_{n,l}$, with $\mu_{n,l}(E) \geq 1 - Cexp(-n^{c_2})$ such that for any $E \in \epsilon$, the following holds: Denote by f_E the density of the random vector $Proj_E(X)$. Then, for any $x \in E$ with $|x| \leq cn^\alpha$,*

$$|\frac{f_E(x)}{\phi(x)} - 1| \leq \frac{C}{n^{c_3}}$$

where, $C, c_1, c_2, c_3, \alpha > 0$ are universal constants. Here, $\phi(x) = (2\pi)^{-l/2} exp(-|x|^2/2)$ is the standard gaussian density in E. (7)

3.2 Extension to Non-Isotropic Random Vectors

When X does not have an isotropic density, then we can still assert that $\mathrm{Proj}_E(x)$ is approximately gaussian for some l dimensional subspace $E \subset \mathbb{R}^n$. Thus we cannot project the data on any direction as it is not isotropic but only along directions where the density concentrates in the lower dimensional subspace. If we assume that our data is a random vector which is not isotropic, we can still use Klartag's estimates however with a scaling of covariance matrix, i.e., if A is the covariance matrix and P is the projection operator, then our effective new projection operator $P' := A^{-1/2}PA^{1/2}$. Thus, we map the general case to Lemma 1 by the change in the projection operator which involves scaling by the covariance matrix. However, the above transformation can blow up errors if we project along directions where the densities don't concentrate.

Generally high dimensional distributions are concentrated around low dimensional subspaces or manifolds. This is particularly true for log-concave distributions. Thus, it make sense to consider only those directions where densities concentrate. Thus, we first perform subspace clustering algorithms on the original data and then project along these directions. Once we get Gaussians on the subspace, we can then learn gaussian mixture models on that space using various methods like those in (8), (9).

4. Problem Statement

With all the necessary tools in hand, we shall now define the problem statement. It is reasonable to assume that data is drawn from a mixture of log-concave densities because they are expressive enough as argued earlier. Following this, we expected that random linear projections of such a vector onto a subspace should be close to a Gaussian Mixture on the lower dimensional subspace. Thus the flow of steps is as follows:

1. It is reasonable to assume that log-concave densities are concentrated in lower dimensional subspaces. We find such directions using subspace clustering and then project the data along these directions to obtain a denser lower dimensional representation.

2. Invoking the result by Klartag and Eldan (7), this lower dimensional representation is expected to be close to a gaussian mixture with the same weights as the original mixture with high probability. Learn a gaussian mixture model in this space using standard algorithms like expectation maximization.

3. Comment on the higher dimensional estimates using the Johnson Lindenstrauss Lemma

The closest to our work is the work by Dasgupta (8), where it is shown that learning a GMM on a higher dimension can be made easier by random projections. It was also shown that this transformation reduced the eccentrity of the high dimensional gaussians which made them easier to learn. The main contribution of this work is considering the case where the multimodal distribution is a more general mixture of log-concave densities as opposed to the simpler case.

5. Theoretical Guarantees

Define $GMM_K(x)$: The pdf of a random vector following a gaussian mixture model with K components and which can be expressed in the form of $\sum_{i=1}^{i=K} w_i \mathcal{N}(x; \mu_i, \Sigma_i)$
Consider a random vector $X \in \mathbb{R}^D$ which follows a mixture of log-concave densities $p(x)$ with K (finite) components subject to $\sum_i w_i = 1$ and each f_i being log concave

$$f_X(x) = \sum_{i=1}^{i=K} w_i f_i(x)$$

Consider a random projection of the data vector onto a subspace of dimensionality $d < D$ by the operator ϕ. Typically, entries of ϕ shall be either sampled from a gaussian or a bernoulli with ±1 values with the columns appropriately scaled. Let $Y_1, Y_2, ..Y_K$ be random variables from the K component distributions respectively. Consider $\varepsilon \subseteq G_{D,d}$, then for **some** $E \subseteq \varepsilon$, $Proj_E(X) = \phi X$ for a fixed ϕ. Then for all $A \subseteq E$ which are measurable we have the following :

$$\mathbb{P}_X(\phi X \in A) = \mathbb{E}_X[\mathbb{I}(\phi X \in A)] = \int \mathbb{I}(\phi X \in A) f_X(x) dx$$

In the above $\mathbb{I}$ denotes the indicator function. Consider a random variable I which takes values $1, 2, ...K$ with $\mathbb{P}(I = i) = w_i$. Using this the above mixture model can be viewed as a latent variable model with the latent variable I and having the following -

$$f_X(x) = \sum_{i=1}^{i=K} P(I = i) f_X(x|I = i)$$

Plugging the above expression in the equation obtained earlier we get -

$$\mathbb{P}_X(\phi X \in A) = \int \mathbb{I}(\phi X \in A) \sum_{i=1}^{i=K} P(I = i) f_X(x|I = i) dx$$

$$= \sum_{i=1}^{i=K} P(I = i) \int \mathbb{I}(\phi X \in A) f_X(x|I = i) dx$$

We shall now look at the integral in the summation, given that we known that X comes from which component of the mixture (as the pdf is conditioned on $I = i$, we can assert the following statement -

$$\int \mathbb{I}(\phi X \in A) f_X(x|I = i) = \mathbb{P}(\phi Y_i \in A)$$

$$\mathbb{P}_X(\phi X \in A) = \sum_{i=1}^{i=K} w_i \mathbb{P}(\phi Y_i \in A)$$

Results of Klartag (6) suggest the following : If $dim(E) < D^c$, for a certain vector Y following a log concave density, then for a certain gaussian random vector Z in the subspace E, and for some constant C, we have the following results:

$$\sup_{A \subseteq E} |\mathbb{P}(Proj_E(Y) \in A) - P(Z \in A)| \leq \frac{C}{D^c}$$

We can use the above for each Y_i as they are drawn for f_i which is a log concave density. Hence for gaussian vectors $Z_1, Z_2, ...Z_K$(with $Z_i \sim \mathcal{N}(\mu_i, \Sigma_i)$), then considering the following gaussian mixture density and a random vector $\Gamma \sim GMM$

$$GMM(x) = \sum_{i=1}^{i=K} w_i \mathcal{N}(x; \mu_i, \Sigma_i)$$

$$\mathbb{P}(\Gamma \in A) = \int_A \sum_{i=1}^{i=K} w_i \mathcal{N}(x; \mu_i, \Sigma_i) = \sum_{i=1}^{i=K} w_i \int_A \mathcal{N}(x; \mu_i, \Sigma_i) = \sum_{i=1}^{i=K} w_i \mathbb{P}(Z_i \in A)$$

$$\sup_{A \subseteq E} |\mathbb{P}_X(\phi X \in A) - \mathbb{P}(\Gamma \in A)| = \sup_{A \subseteq E} |\sum_{i=1}^{i=K} w_i \mathbb{P}(\phi Y_i \in A) - \sum_{i=1}^{i=K} w_i \mathbb{P}(Z_i \in A)|$$

$$\leq \sum_{i=1}^{i=K} w_i \sup_{A \subseteq E} |\mathbb{P}(\phi Y_i \in A) - \mathbb{P}(Z_i \in A)| \leq \sum_{i=1}^{i=K} w_i \frac{C_i}{D^c} = \frac{C'}{D^c}$$

Here $C' = \sum_{i=1}^{i=K} w_i C_i$ which is constant. Thus the density of a random lower dimensional projections of log concave mixtures are close to a gaussian mixture in a total variation sense.

6. Sparse Subspace Clustering

6.1 Problem Formulation

As motivated earlier, we need to identify the directions where densities concentrate which are believed to be in subspace. This chapter defines the problem of subspace clustering and various algorithms the optimization problem which arises as a result of the formulation.

Many real-world problems deal with collections of high-dimensional data, such as images, videos, text and web documents, DNA microarray data, and more. Often, such high-dimensional data lie close to low-dimensional structures corresponding to several classes or categories to which the data belong. The key idea is that, among the infinitely many possible representations of a data point in terms of other points, a sparse representation corresponds to selecting a few points from the same subspace. We now look into certain algorithms that are used for subspace clustering on affine spaces.

6.2 k-Means Clustering

This is the baseline algorithm used for clustering of points based on euclidean distance and not on the basis of them belonging to a common subspace. We shall use this as the baseline algorithm to compare other algorithms for subspace clustering. k-Means clustering is clearly not desirable because it shall mess up the points at the intersection of subspaces and won't cover far away points which still lie on the same hyperplane.

6.3 Spectral Clustering for SSC

This work by You et.al. (10) was one of the earliest works in subspace clustering for affine spaces. Consider we have a data matrix $X \in \mathbb{R}^{D \times N}$ where N is the number of samples and D is the dimensionality of the data. We use self-representation, i.e., for a particular datapoint x_i, we can write it as a linear combination of points belonging in the same subspace. Thus we aim to solve :

$$C^* = \text{argmin } ||C||_1, \text{s.t. } X = XC \text{ and } \text{diag}(C) = 0$$

where $C \in \mathbb{R}^{N \times N}$ is the representation matrix and we want the diagonal elements to be 0 in order to avoid the trivial solution of each datapoint being represented as itself and with 0 weights given to other. We then define the matrix $W = C + C^T$ and then use spectral clustering techniques on W to get clusters of the data. It has been shown that under mild condition, SSC is subspace preserving, i.e., if $c_{i,j} = 0$ implies x_i and x_j belong to different subspaces.

6.4 Elastic Net Subspace Clustering

In (11), instead of minimizing just the l-1 norm of the connectivity matrix, we penalize the weighted sum of the l-1 and l-2 penalities with a trade off parameter λ. Thus the loss function which we aim to optimize is:

$$c_i^* = \text{argmin } \lambda||c_i||_1 + (1-\lambda)||c_i||_2^2, \text{s.t. } x_i = Xc_i \text{ and } c_i = 0$$

6.5 Orthogonal Matching Pursuit for SSC

To scale the SSC algorithm to large datasets, (12) using OMP like updates as in compressed sensing to find the connectivity matrix for solving the same optimization problem as in Spectral Clustering. There is still a difference of using OMP for subspace clustering as the recovered matrix need not be unique unlike signal recovery in the case of compressed sensing. (12) shows some theoretical bounds as well for subspace identification and is shown to be scalable to upto $100,000$ data points which is most relevant for this work.

6.6 Empirical Results

We present some results for each of the above algorithm on synthetically generated dataset
Experimental Conditions :
Ambient Dimension : 100
Subspace Dimension : 15
Number of Subspaces in the Union : 10
Number of Samples : 10,000
Results :

Algorithm	Clustering Accuracy
k-means	33.2
Spectral Clustering	65.6
Elastic Net	89.3
SSC-OMP	92.6

Table 6.1: Basic comparision of various subspace clustering algorithms

SSC-OMP is used for all the subsequent experiments to find the directions where the component log-concave densities concentrate. Once clustered in the high dimensional space, each cluster is projected at appropriate directions.

7. Learning Mixture of Gaussians

Once we have points in the projected space, we can learn the parameters of a gaussian mixture model using standard methods like expectation maximization. To recall, a GMM with k components is a parameterised density estimator taking the form:

$$p_X(x) = \sum_{i=1}^{i=K} w_i \mathcal{N}(x; \mu_i, \Sigma_i) = \sum_{z} p(z)p(x|z)$$

We start with random initialization of these parameters and then iterate over the following two steps until convergence. We check the negative log-likelihood values as the metric needed to be minimised and stop the iterations when the relative change falls below some ϵ which is a hyperparameter.

1. **E-step**: Compute the posterior probability over z given our current model - i.e. how much do we think each Gaussian generates each datapoint.

   $$\gamma_k = p(z = k|x) = \frac{p(z = k)p(x|z = k)}{p(x)} = \frac{w_k \mathcal{N}(x; \mu_k, \Sigma_k)}{\sum_{i=1}^{i=K} w_i \mathcal{N}(x; \mu_i, \Sigma_i)}$$

   γ_k can be viewed as the responsibility of cluster k towards x

2. **M-step**: Assuming that the data really was generated this way, change the parameters of each Gaussian to maximize the probability that it would generate the data it is currently responsible for. The re-estimation of parameters is done in the following way -

   $$\mu_k = \frac{\sum_{i=1}^{i=N} \gamma_k^{(n)} x^{(n)}}{N_k}, \Sigma_k = \frac{\sum_{i=1}^{i=N} \gamma_k^{(n)} (x^{(n)} - \mu_k)(x^{(n)} - \mu_k)^T}{N_k}$$

   $$w_k = \frac{N_k}{N}, N_k = \sum_{n=1}^{n=N} \gamma_k^{(n)}$$

3. Evaluate log likelihood and check for convergence

   $$\log p(x|w, \mu, \Sigma) = \sum_{n=1}^{n=N} \log \sum_{i=1}^{i=K} w_i \mathcal{N}(x; \mu_i, \Sigma_i)$$

8. Preliminary Experiments

For a simple experiment, we take $D = 5$ and take three log-concave distributions - a multivariate gaussian, and 2 laplace distributions with different means and decaying factors. The 3 component means were taken to be well away from each other without much overlap. A total of 20000 samples were drawn from the mixture with weights $[0.2, 0.5, 0.3]$ and a linear operator ϕ with bernoulli entries was used for projection on a 2D space(so that we can visualize). Following are the scatter plots of the random projections obtained on 2 such ϕ instances.

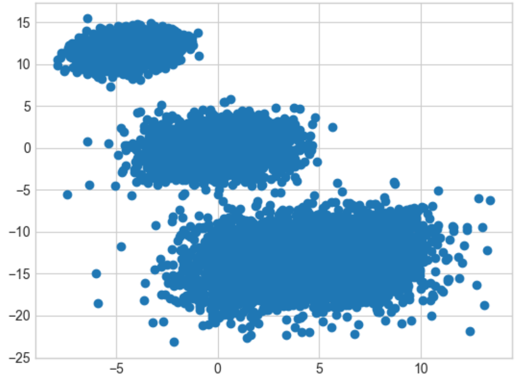


Figure 8.1: Low dimensional projections for ϕ_1

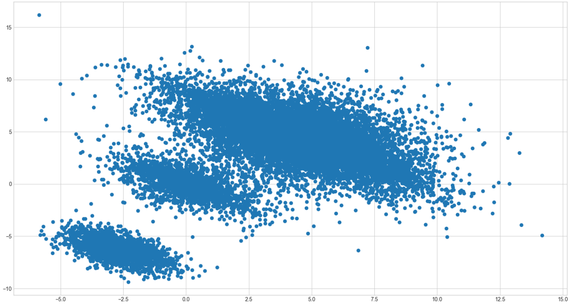


Figure 8.2: Low dimensional projections for ϕ_2

The above preliminary empirical experiments suggest clusters are preserved in the lower dimensional space if they are far apart originally which makes them suitable for learning Gaussian mixture models.

9. Empirical Results

9.1 Synthetic Data

Here, we generate data from randomly sampling from a mixture of log concave densities and then fit a GMM on the projected space [no clustering yet]. We hope to recover the weights upto a permutation ambiguity. Further we hope to get the density well only if we have $O(\frac{1}{\min w_i})$, i.e, enough samples even to represent the least weighted cluster so that those points do not get subsumed in some other cluster or be considered as outliers. Consider the following distributions and how weights are preserved in them -

Example 1 :

Ambient dimension D : 50

Projected dimension d : 20

Mixture Components k : 4

The 4 components were a Gaussian, Laplacian, Beta and Uniform distribution. The elements of ϕ were randomly sampled from a standard normal gaussian. Number of samples were 10k. For each row of the table below, the parameters of the above distributions and the entries of ϕ were generated randomly.

Original Mixture Weights	Weights estimated by the GMM Algorithm
0.25, 0.25, 0.25, 0.25	0.249, 0.248, 0.249, 0.252
0.15, 0.5, 0.14, 0.2	0.156, 0.498, 0.146, 0.199
0.4, 0.3, 0.2, 0.1	0.401, 0.295, 0.202, 0.1
0.1, 0.7, 0.1, 0.1	0.1, 0.699, 0.1, 0.1

Table 9.1: Comparison of weights of the mixture with those predicted from the GMM

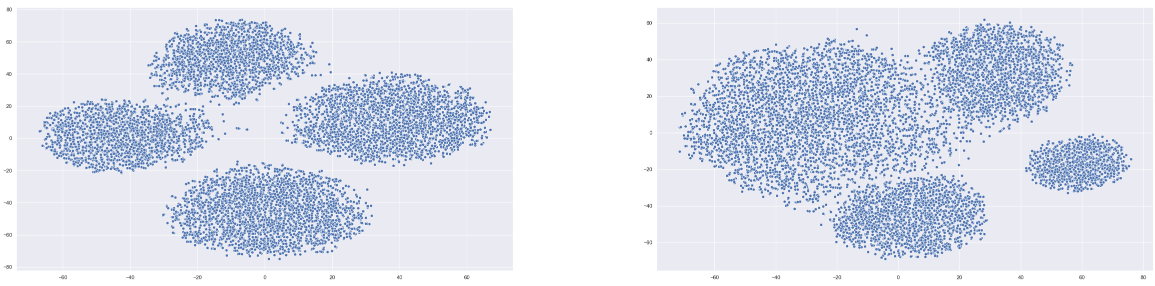


Figure 9.1: t-SNE embeddings on 2d for the 15d space which indicates 4 clusters which should be present in the higher dimension as well

Example 2:
Ambient dimension D : 150
Projected dimension d : 50
Mixture Components k : 3/6
The 3/6 components were a Gaussian, Laplacian Uniform distribution. The elements of ϕ were randomly sampled from a standard normal gaussian. Number of samples were 10k. For each row of the table below, the parameters of the above distributions and the entries of ϕ were generated randomly.

Original Mixture Weights	Weights estimated by the GMM Algorithm
0.173, 0.48, 0.346	0.172, 0.364, 0.46
0.658, 0.163, 0.179	0.655, 0.165, 0.179
0.218, 0.042, 0.262, 0.176, 0.194, 0.107	0.217, 0.051, 0.305, 0.176, 0.193, 0.057
0.213, 0.044, 0.253, 0.139, 0.193, 0.156	0.214, 0.077, 0.297, 0.139, 0.194, 0.082

Table 9.2: Comparison of weights of the mixture with those predicted from the GMM

9.2 Real Data

9.2.1 MNIST Dataset

Using Raw Data for Projections:

MNIST Images which were 784 dimensional were projected onto a 500 dimensional space using a random Gaussian matrix with entries iid for a normal gaussian. A GMM with 10 components was fitted on the projected space and we measure the log-likelihoods of the same on both the train and the test split. The train split had 60k images while the test split had 10k images. We also 'decode' after sampling from the GMM by using basis pursuit as mnist images are sparse in their canonical basis. Following are the results of the same :
Train Log Likelihood: -206.12
Test Log Likelihood: -211.47

Basis Pursuit -

$$y = \Phi x + \eta$$

$$\text{recovery using the following : } \min \|x\|_1 \text{s.t.} \|y - \Phi x\|_2 \leq \epsilon$$

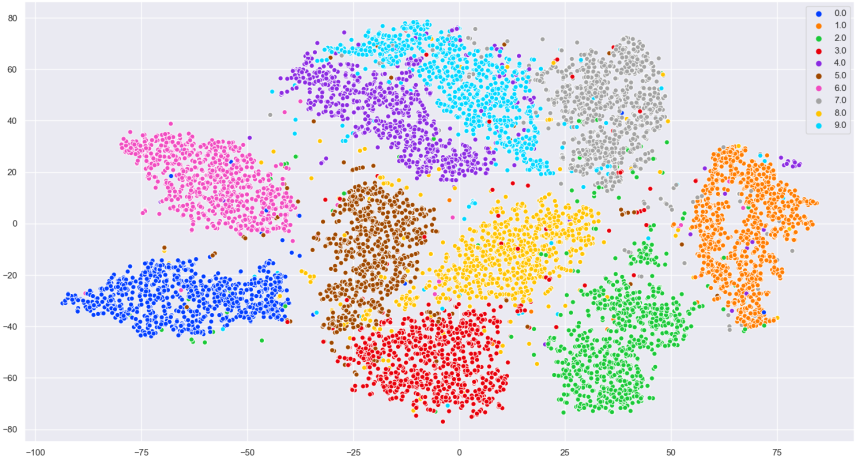


Figure 9.2: t-SNE embeddings of the projected space of the test samples where each color code indicates a digit class

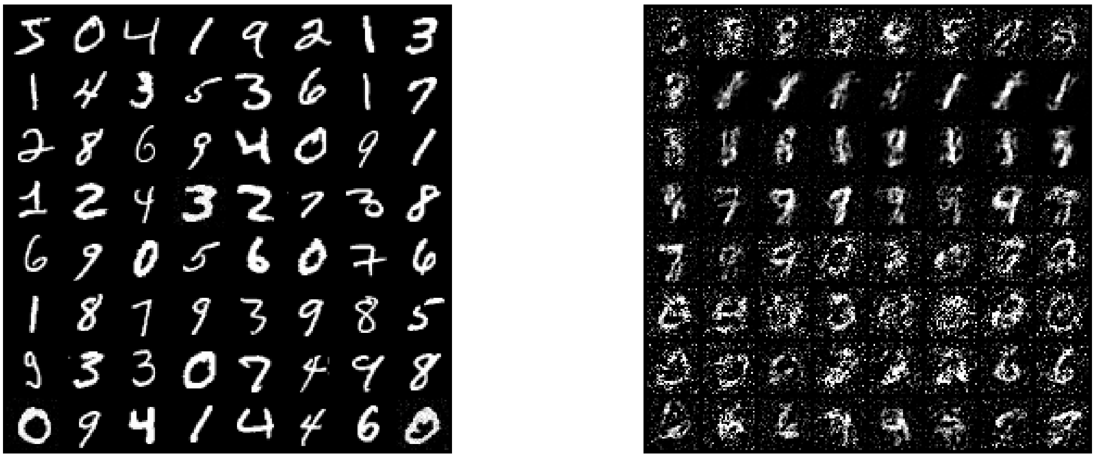


Figure 9.3: Groundtruth train image after CS recovery (left) and samples drawn from the fitted GMM after decoding (right)

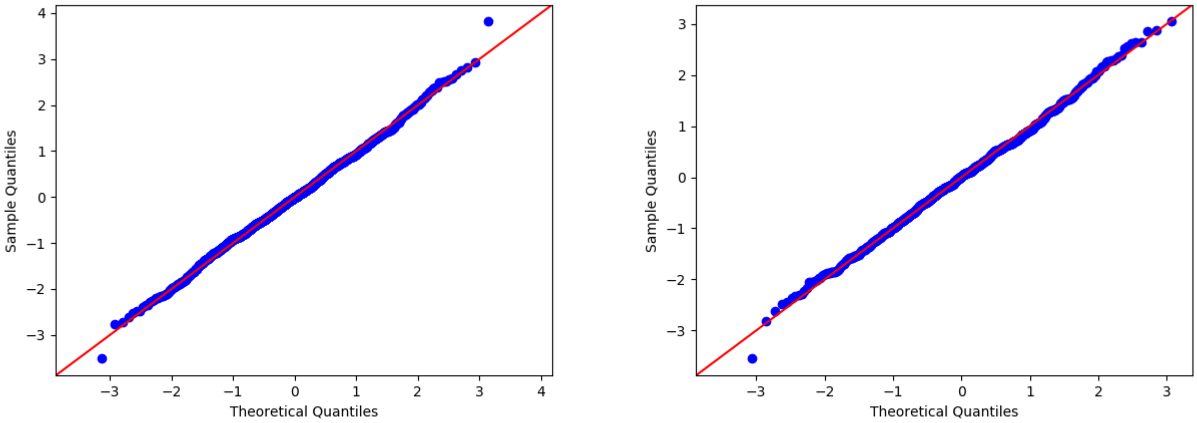
Projections after PCA:

In this part of the experiment, we first take the PCA projections of the images(top 200) and project them on a space of size 50 using the similar Gaussian matrix. We then fit a GMM with 10 components on this 50 dimensional space.

Train Log Likelihood: -140.908

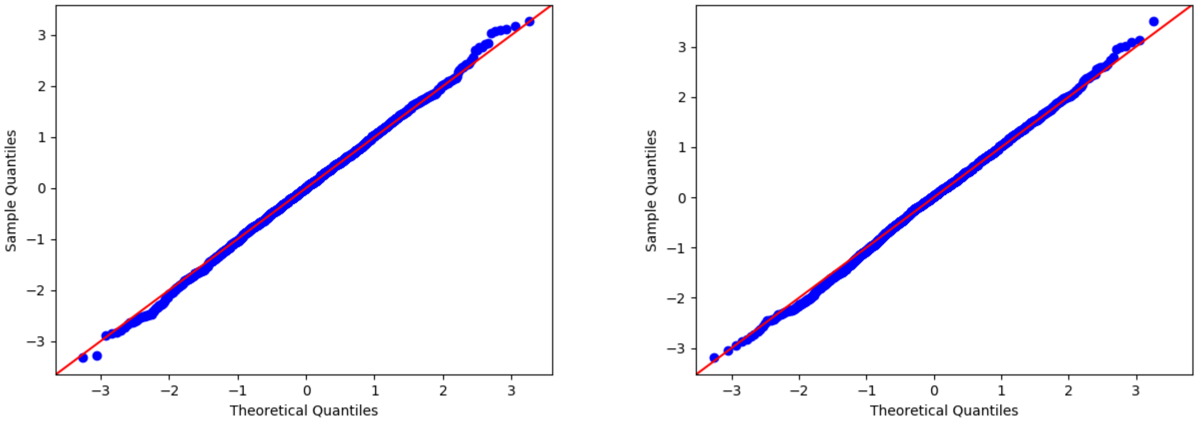
Test Log Likelihood: -140.88

To compare whether the samples in the lower dimensional subspace are indeed Gaussian, we use the help of Q-Q plots. More specifically, we first fix a label l and fetch all the points which have been assigned that label l by the GMM. These points are then whitened by an affine transformation, more specifically, $X' := \Sigma^{-1/2}(X - \mu)$ where μ, Σ is the mean and covariance matrix respectively. We compare these set of points X' with a standard normal Gaussian in terms of quantiles. A straight line with a slope of 1 indicates that the distributions plotted along the axes being exactly equal to each other.



(a) Q-Q plot for label 0

(b) Q-Q plot for label 4



(a) Q-Q plot for label 7

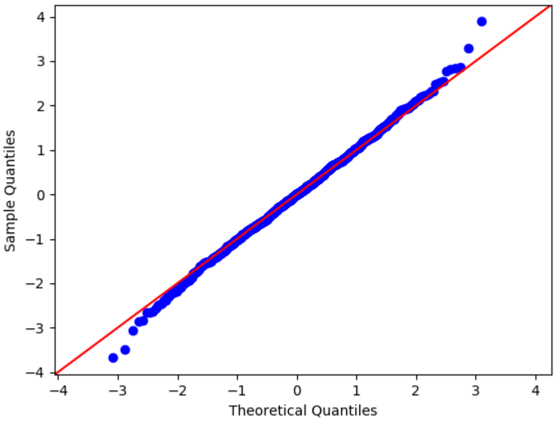
(b) Q-Q plot for label 9

9.2.2 Labeled Faces in the Wild Dataset

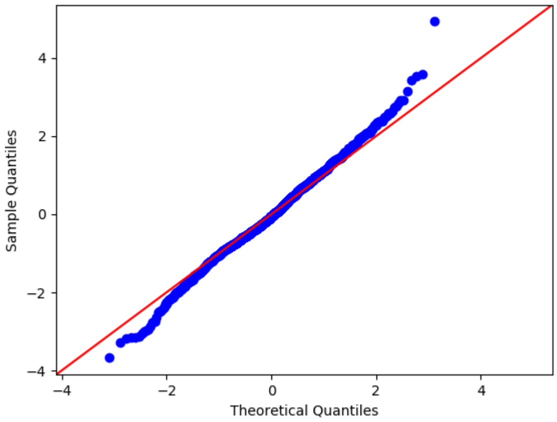
This dataset has a total of 13233 samples of human faces out of which 10k samples are in the train split while the other in the test split. Each samples had a dimensionality of 5828 with values being floating point numbers between 0 to 255. PCA was performed on this raw data and top 700 eigen values were taken for transforming the data. This transformed data was projected onto a 300-d space for fitting a gaussian mixture.

Train Log Likelihood: -2451.13
Test Log Likelihood: -2457.64

As in the case of MNIST, we look at the Q-Q plot for datapoints given the same label by the GMM algorithm to compare whether it mimics a standard gaussian after an affine tranformation of mean subtration and scaling by $\Sigma^{-1/2}$.



(a) Q-Q plot for label 0

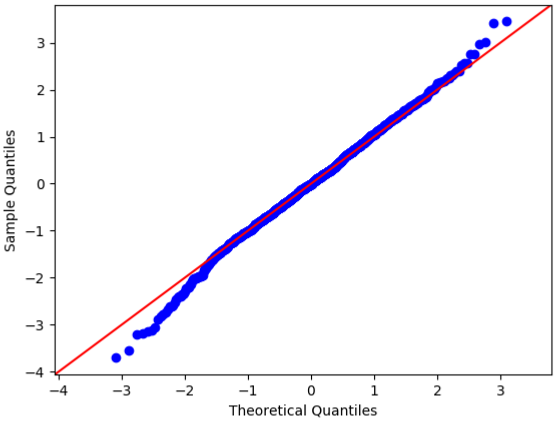


(b) Q-Q plot for label 1

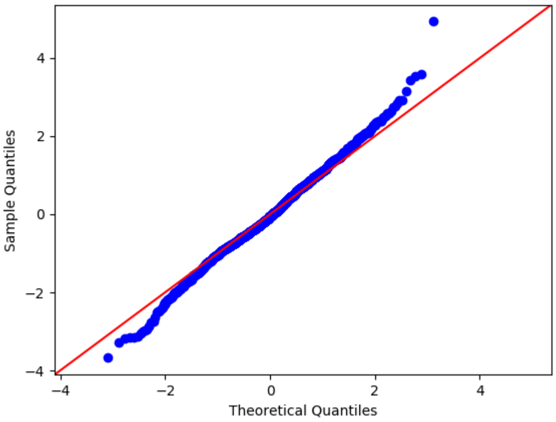
The slope 1 line in Q-Q plots corroborate our claim that random images of mixture of log-concave densities is indeed a gaussian mixture. The difference at the tails of the distribution can be expected as even theoretically, we just guarantee closeness in a total variation sense and not a pointwise equality. As anyways the tail probabilities are less, they can be different for both the distributions while still allowing us to bound the total variation difference. Further we can always have outliers which can also affect the GMM learning algorithm which is not robust.

9.3 Subspace Clustering Post Projection

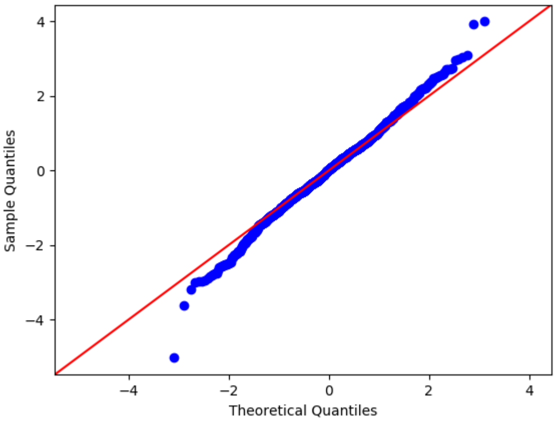
Random projection is essentially a linear map which shall preserve the structure of data lying in a union of subspaces. To save computational cost, we perform subspace clustering post projection and then fit a gaussian mixture model. The experimental conditions were the same for subspace clustering before projection. The Q-Q plots for the above two datasets are presented below -



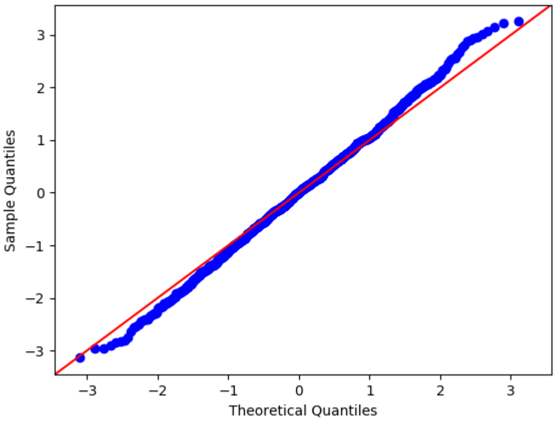
(a) Q-Q plot for MNIST label 2



(b) Q-Q plot for MNIST label 6



(a) Q-Q plot for Faces label 3



(b) Q-Q plot for Faces label 6

9.4 Choice of Projection Dimension and other Parameters

There were two factors which governed the choice of the dimension of subspace for projection:

1. Subspace clustering using SSC-OMP requires just the number of clusters in the union as the input parameter which was set equal to the number of gaussians in the mixture.

2. The latent dimension d should be of the order $\mathcal{O}(\log N/\epsilon^2)$ for us to invoke the JL lemma where N is the number of the datapoints in the higher dimensional space.

3. For the MNIST experiment specifically, the latent dimension was chosen to be a bit higher (500) so as to enable compressed sensing recovery by sampling from the gaussian mixture model fitted on the latent space and visualize the samples.

10. Invoking the JL Lemma

Once we have fit a GMM on the subspaces, then we can invoke the JL Lemma to comment on the estimates of various moments in the higher dimension. As shown earlier, the mixture weights are preserved with high probability. Thus the projected points of each cluster in the gaussian mixture correspond to a log-concave density in the higher dimension. Given we know all the moments in the lower dimension, we can invoke the JL Lemma to comment about higher dimensional estimates. This is similar to the mathematical analysis done for mapping of estimates in (8). Once we have the means in the higher dimension, then we can estimate the second moment as well. Say indices belonging to a set A form a cluster and $|A| = N$, then covariance matrix for that cluster can be estimated

$$\Sigma_k = \sum_{i \in A} \frac{(X_i - \mu_k)(X_i - \mu_k)^T}{N}$$

Given that the lower dimensional projections are gaussian, the higher moments carry no extra information in the sense that they will only be functions of the first two moments. Further, even if we have all the moments of a log-concave density, we cannot reconstruct the true pdf without any extra assumptions (13). This is the moment problem. Thus using the method of random projections for density estimation, we get the best fit gaussian for each log-concave density in the higher dimension using the maximum entropy principle(given the mean and variance of a distribution, gaussian maximises the entropy). The component weights can be retrieved exactly with high probability as shown in the theoretical and empirical results.

11. Conclusion and Future Work

We present an easy to implement method to for estimating multimodal high dimensional densities which can be expressed as a mixture of log-concave distributions. We use random projections as a way to reduce to dimensionality and show that this projected data is very close to a gaussian mixture. This claim of ours is corroborated by experiments on both synthetic data and a couple of real data sets. The JL Lemma can be invoked thereafter to comment on higher dimensional estimates. Some lines of future work can be :

- Whether the assumption of mixture of log-concave densities can be relaxed to a more general distribution involving a combination of sum and product nodes?
- Whether a non-linear transformation $f(.)$ which preserves distances like a random matrix multiplication also induces a structure like a GMM on the projections? This would be helpful in analysing the latent spaces of current deep neural networks.

Bibliography

[1] K. P. Murphy, Machine learning: a probabilistic perspective. MIT press, 2012.

[2] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola and L. K. Saul. An Introduction to Variational Methods for Graphical Models. M. I. Jordan (Ed.), Learning in Graphical Models. Kluwer, Dordrecht, The Netherlands, 1998.

[3] Jianguo Ding, Probabilistic Inferences in Bayesian Networks, https://arxiv.org/pdf/1011.0935.pdf

[4] Reynolds D. (2009) Gaussian Mixture Models. In: Li S.Z., Jain A. (eds) Encyclopedia of Biometrics. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-73003-5_196

[5] S. Dasgupta and A. Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. Random Structures and Algorithms, 22(1):60-65, 2003.

[6] A central limit theorem for convex sets - B. Klartag, Invent. Math., Vol. 168, 91-131

[7] Pointwise Estimates for Marginals of Convex Bodies - B. Klartag, R. Eldan, J. Functional Analysis, Vol. 254, Issue 8, (2008), 2275-2293

[8] S. Dasgupta. Learning mixtures of Gaussians. Fortieth Annual IEEE Symposium on Foundations of Computer Science (FOCS), 1999.

[9] Settling the Polynomial Learnability of Mixtures of Gaussians, Ankur Moitra, Greg Valiant - Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2010)

[10] Sparse Subspace Clustering: Algorithm, Theory, and Applications - Ehsan Elhamifar and René Vidal, IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(11):27652781, 2013.

[11] Oracle Based Active Set Algorithm for Scalable Elastic Net Subspace Clustering - C. You, C.-G. Li, D. Robinson, and R. Vidal, Arxiv, 2016.

[12] Scalable Sparse Subspace Clustering by Orthogonal Matching Pursuit - C. You, D. Robinson, and R. Vidal, Arxiv, abs/1507.01238, 2015.

[13] Alexandros Eskenazis, Piotr Nayar, Tomasz Tkocz, Sharp comparison of moments and the log-concave moment problem, Advances in Mathematics, Volume 334, 2018, Pages 389-416, ISSN 0001-8708, https://doi.org/10.1016/j.aim.2018.06.014.